NEURAL INFORMATION PROCESSING SYSTEMS

MESA: Boost Ensemble Imbalanced Learning with MEta-SAmpler

Zhining Liu, Pengfei Wei, Jing Jiang, Wei Cao, Jiang Bian, and Yi Chang

in 34th Conference on Neural Information Processing Systems (NeurIPS 2020)

MOTIVATION

☐ Problem:

- Inconsistency between:
  - Class-imbalanced data representation
  - Class-balanced accuracy-oriented learning process
- **Goal: learning unbiased models from class-imbalanced data**

☐ Limitations of Existing Work:

- **The assumptions they made on the data may not hold, resulting in:**
  - Unstable performance due to the sensitivity to outliers
  - High cost of computing the distance between instances.
  - Poor applicability because of the prerequisite of domain experts to hand-craft the cost matrix

☐ Comparisons of MESA with existing imbalanced learning methods:

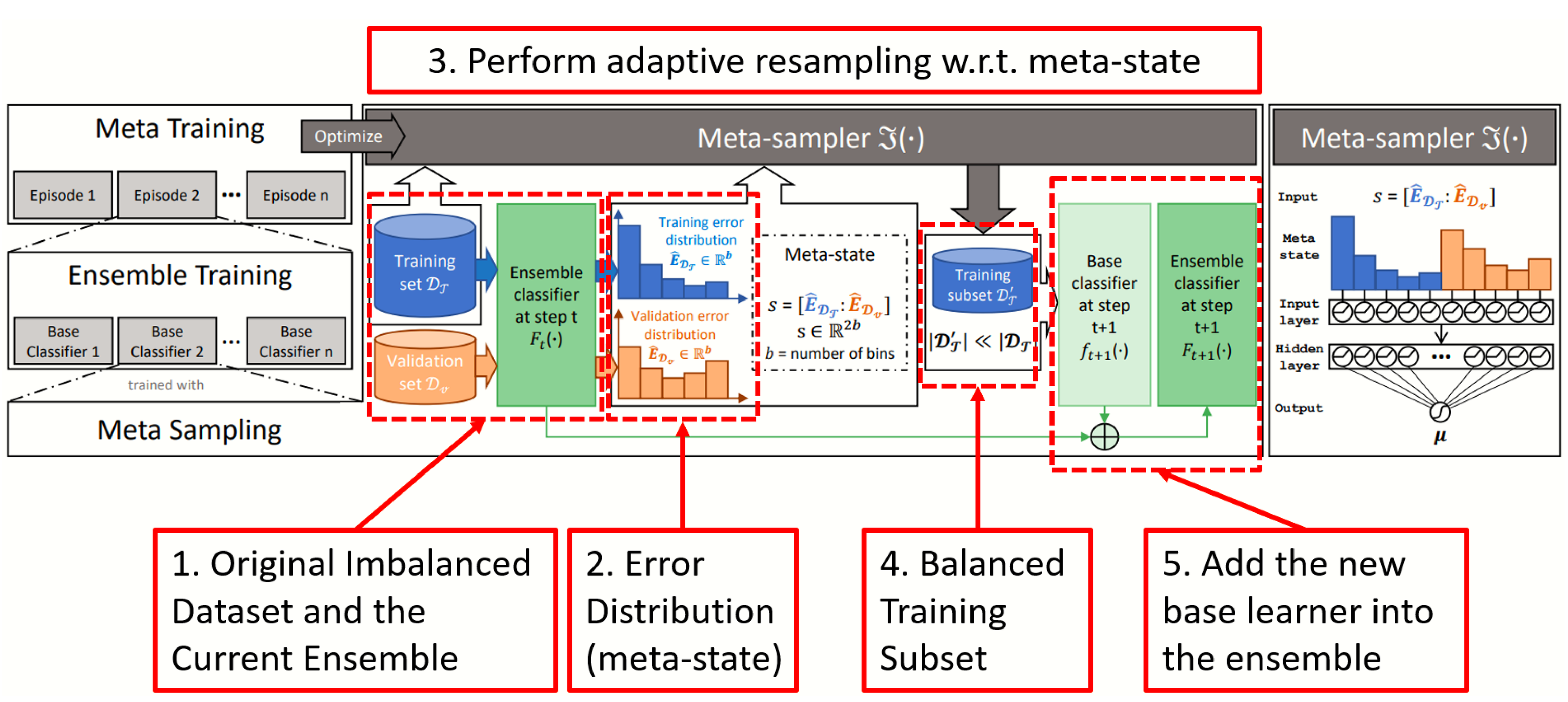
Category*	Representative(s)	Sample efficiency	Distance-based resampling cost	Domain knowledge free?	Robust to noises/outliers?	Requirements
RW	[31], [5]	$\mathcal{O}(\lvert\mathcal{P}\rvert+\lvert\mathcal{N}\rvert)$	✗	✗	✗	cost matrix set by domain experts
US	[35], [42]	$\mathcal{O}(2\lvert\mathcal{P}\rvert)$	$\mathcal{O}(\lvert\mathcal{P}\rvert)$	✓	✗	well-defined distance metric
OS	[6], [17]	$\mathcal{O}(2\lvert\mathcal{N}\rvert)$	$\mathcal{O}(\lvert\mathcal{P}\rvert)$	✓	✗	well-defined distance metric
CS	[47], [44]	$\mathcal{O}(\lvert\mathcal{P}\rvert+\lvert\mathcal{N}\rvert)$	$\mathcal{O}(\lvert\mathcal{P}\rvert\cdot\lvert\mathcal{N}\rvert)$	✓	✓	well-defined distance metric
OS+CS	[4], [3]	$\mathcal{O}(2\lvert\mathcal{N}\rvert)$	$\mathcal{O}(\lvert\mathcal{P}\rvert\cdot\lvert\mathcal{N}\rvert)$	✓	✓	well-defined distance metric
IE+RW	[12], [43]	$\mathcal{O}(k(\lvert\mathcal{P}\rvert+\lvert\mathcal{N}\rvert))$	✗	✗	✗	cost matrix set by domain experts
PE+US	[2], [32]	$\mathcal{O}(2k\lvert\mathcal{P}\rvert)$	✗	✓	✓	-
PE+OS	[46]	$\mathcal{O}(2k\lvert\mathcal{N}\rvert)$	$\mathcal{O}(2k\lvert\mathcal{P}\rvert)$	✓	✓	well-defined distance metric
IE+RW+US	[39]	$\mathcal{O}(2k\lvert\mathcal{P}\rvert)$	✗	✓	✗	-
IE+RW+OS	[7]	$\mathcal{O}(2k\lvert\mathcal{N}\rvert)$	$\mathcal{O}(2k\lvert\mathcal{P}\rvert)$	✓	✗	well-defined distance metric
ML	[41], [38], [48]	$\mathcal{O}(\lvert\mathcal{P}\rvert+\lvert\mathcal{N}\rvert)$	✗	✗	✓	co-optimized with DNN only
IE+ML	MESA(ours)	$\mathcal{O}(2k\lvert\mathcal{P}\rvert)$	✗	✓	✓	independent meta-training

* reweighting (RW), under-sampling (US), over-sampling (OS), cleaning-sampling (CS), iterative ensemble (IE), parallel ensemble (PE), meta-learning (ML).

THE PROPOSED MESA FRAMEWORK

☐ Overview of the proposed MESA Framework.

- We introduce a novel ensemble imbalanced learning (EIL) framework named MESA. It adaptively resamples the training set in iterations to get multiple classifiers and forms a cascade ensemble model. MESA directly learns a parameterized sampling strategy (i.e., meta-sampler) from data to optimize the final metric beyond following random heuristics.
- It consists of three parts: **meta sampling** as well as **ensemble training** to build ensemble classifiers, and **meta-training** to optimize the meta-sampler.



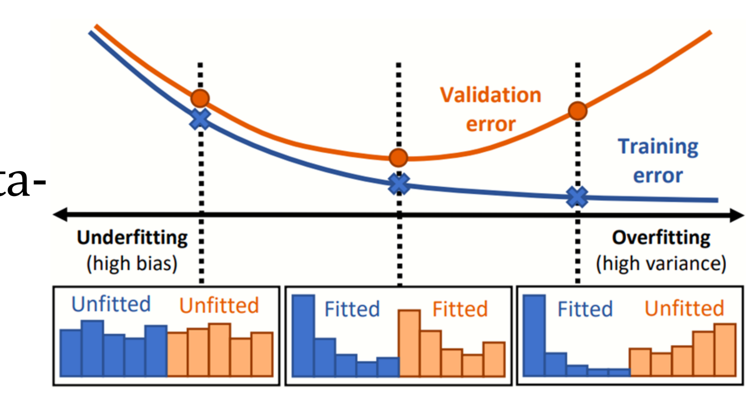
☐ Main features of MESA.

- **Better performance.** Perform adaptive resampling based on meta-information to further boost the performance of ensemble classifiers;
- **Wide applicability.** Decouple model-training and meta-training for general applicability to different classifiers;
- **Transferability.** Train the meta-sampler over task-agnostic meta-data for cross-task transferability and reducing meta-training cost on new tasks.

☐ Meta-state.

- **Histogram distribution of prediction error.**

It shows the distribution of "easy" and "hard" samples in finer granularity and provides the meta-sampler with information about bias/variance of the classifier and thus supporting its decision.



- See an example in the right figure.

☐ Meta-sampling.

To prevent the usage of complex sampler model architecture, we use a **Gaussian function trick** to **simplify the meta-sampling process and the sampler itself.** The meta-sampler outputs a scalar $\mu \in [0, 1]$ based on the input meta-state, we then apply a Gaussian function $g_{\mu,\sigma}(x)$ over each instance's classification error to decide its (unnormalized) sampling weight, where $g_{\mu,\sigma}(x)$ is defined as:

$$g_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}.$$

Note that e is the Euler's number, $\mu \in [0, 1]$ is given by the meta-sampler and σ is a hyperparameter. For detailed discussions about this hyper-parameter setting, please see the appendix provided in the supplementary file.

☐ Ensemble Training.

Given a meta-sampler, we can **iteratively train new base classifiers using the dataset sampled by the sampler**. Please see the process in the figure on the left.

☐ Meta Training.

The meta-sampler is expected to learn and adapt its strategy from the state(s)-action(μ)-state(new s) interactions in the ensemble training process. This meta-training problem can be naturally approached via **reinforcement learning**.

Action: *μ (the resampling parameter, meta-sampler's output)*
Reward: *Δ generalization performance*
(before and after an update, estimated using the validation set)
State: *error distribution (on both training and validation sets)*

EXPERIMENTAL RESULTS

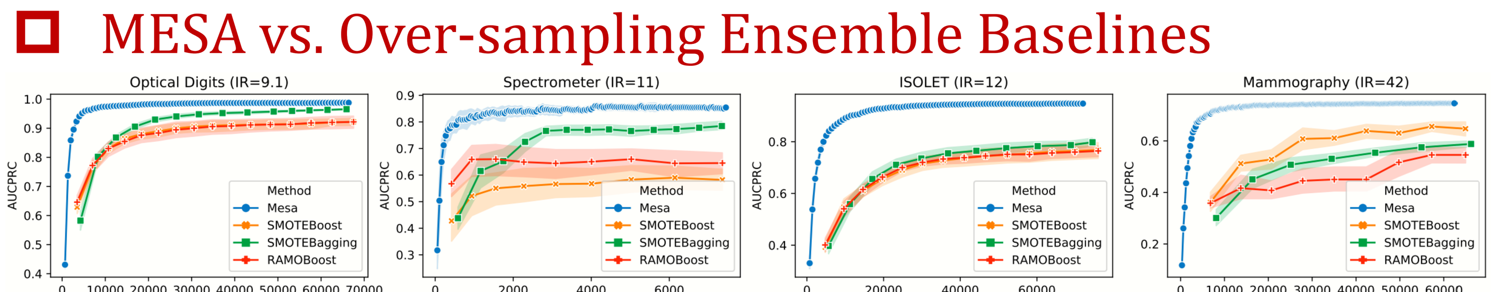
☐ MESA vs. Resampling Baselines

Category	Method	Protein Homo. (IR=111)					#Training Samples	Resampling Time (s)
		KNN	GNB	DT	Boost	GBM		
No resampling	-	0.466	0.742	0.531	0.778	0.796	87,450	-
Under-sampling	RANDOMUS	0.146	0.738	0.071	0.698	0.756	1,554	0.068
	NEARMISS [35]	0.009	0.012	0.012	0.400	0.266	1,554	3.949
Cleaing-sampling	CLEAN [26]	0.469	0.744	0.488	0.781	0.811	86,196	117.739
	ENN [47]	0.460	0.744	0.532	0.789	0.817	86,770	120.046
	TOMEKLINK [45]	0.466	0.743	0.524	0.778	0.791	87,368	90.633
	ALLKNN [44]	0.459	0.744	0.542	0.789	0.816	86,725	327.110
	OSS [24]	0.466	0.743	0.536	0.778	0.789	87,146	92.234
Over-sampling	RANDOMOS	0.335	0.706	0.505	0.736	0.733	173,346	0.098
	SMOTE [6]	0.189	0.753	0.304	0.700	0.719	173,346	0.576
	ADASYN [17]	0.171	0.679	0.315	0.717	0.693	173,366	2.855
	BORDERSMOTE [16]	0.327	0.743	0.448	0.795	0.711	173,346	2.751
Over-sampling + Cleaning	SMOTEENN [4]	0.156	0.750	0.308	0.711	0.750	169,797	156.641
	SMOTETOMEK [3]	0.185	0.749	0.292	0.782	0.703	173,346	116.401
Meta-sampler	MESA (OURS, k=10)	**0.585**	**0.804**	**0.832**	**0.849**	**0.855**	*1,554×10*	*0.235×10*

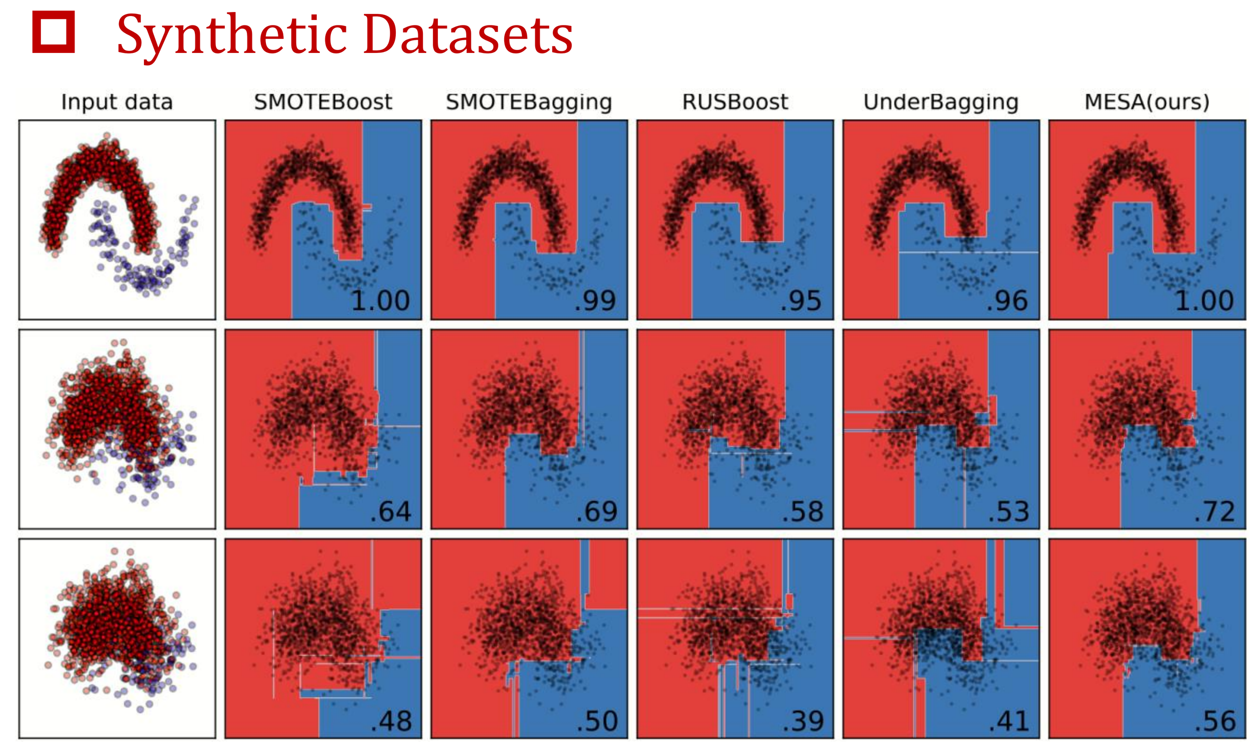
☐ MESA vs. Under-sampling Ensemble Baselines

Method	Optical Digits (IR=9.1)			Spectrometer (IR=11)			ISOLET (IR=12)			Mammography (IR=42)		
	k=5	k=10	k=20	k=5	k=10	k=20	k=5	k=10	k=20	k=5	k=10	k=20
RUSBOOST [39]	0.883	0.946	0.958	0.686	0.784	0.786	0.696	0.770	0.789	0.348	0.511	0.588
UNDERBAGGING [2]	0.876	0.927	0.954	0.610	0.689	0.743	0.688	0.768	0.812	0.307	0.401	0.483
SPE [34]	0.906	0.959	0.969	0.688	0.777	0.803	0.755	0.841	0.895	0.413	0.559	0.664
CASCADE [32]	0.862	0.932	0.958	0.599	0.754	0.789	0.684	0.819	0.891	0.404	0.575	0.670
MESA (OURS)	**0.929**	**0.968**	**0.980**	**0.723**	**0.803**	**0.845**	**0.787**	**0.877**	**0.921**	**0.515**	**0.644**	**0.705**

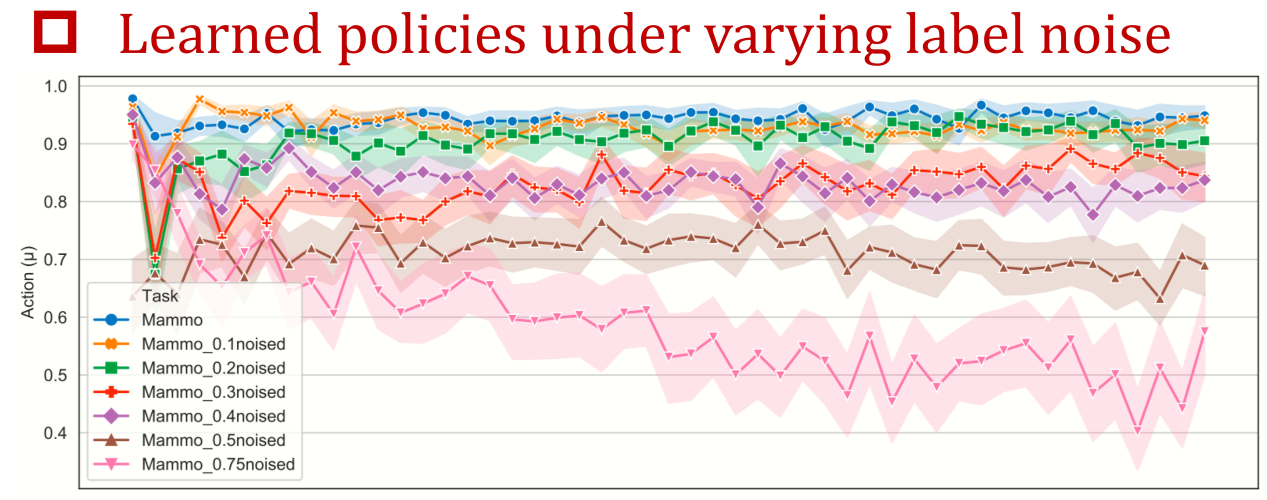
☐ MESA vs. Over-sampling Ensemble Baselines



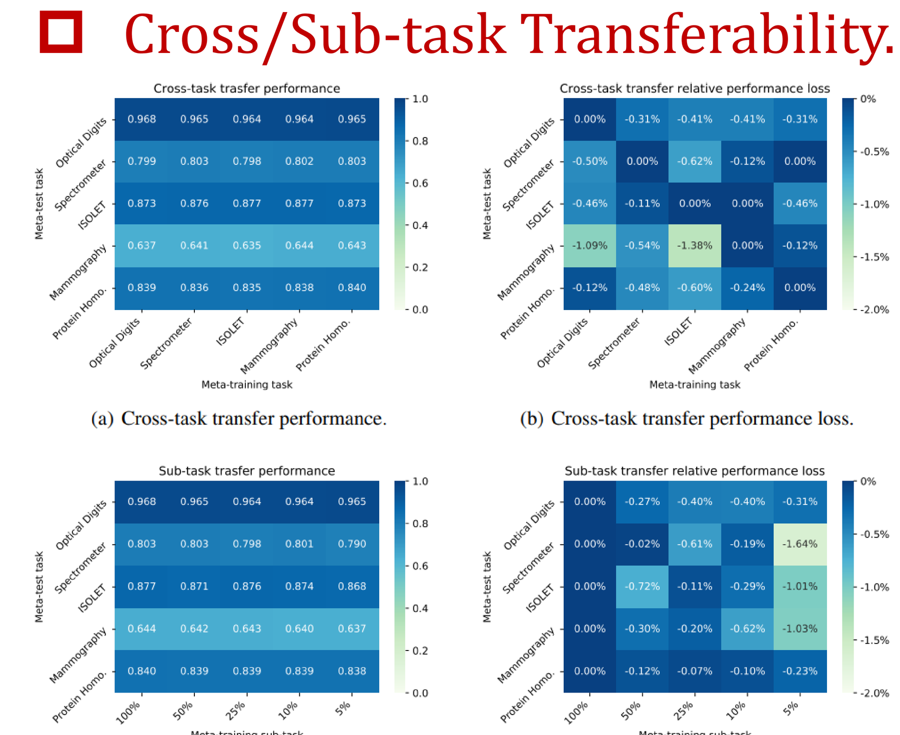
☐ Synthetic Datasets



☐ Learned policies under varying label noise



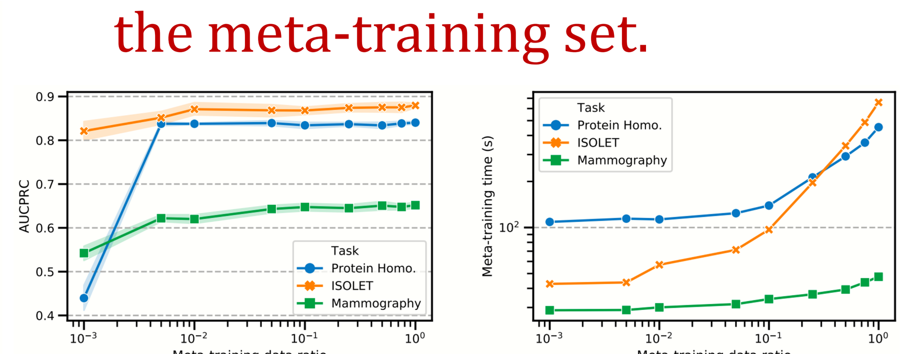
☐ Cross/Sub-task Transferability.



(a) Cross-task transfer performance. (b) Cross-task transfer performance loss.

(c) Sub-task transfer performance. (d) Sub-task transfer performance loss.

☐ The influence of scaling down the meta-training set.



(a) Sub-task transfer performance. (b) Sub-task meta-training time.

Code link: *https://github.com/ZhiningLiu1998/mesa*